

**68000 Sound Driver
Ver. 3.00**

Rex Sabio
242

Introduction

The 68000 Sound Driver Version 3.00 (hereinafter "V3") is designed to fully exploit the sound source that is incorporated into the SUPER32X.

Sound Driver Specifications

The following are the specifications for the Sound Driver.

| | |
|---|---|
| Target sound sources | PSG or 6-sound/PSG tone 3+ notes (UPM4 4-sound PCM is using 8-bit linear or 10-bit 4-bit, 8-bit, 16-bit) |
| Required resources Driver size (68000) Work size CPU load (68000) I/O Music-playing SFC (per channel) CPU load (SFC) | Approximately 1000H (including a 280 driver (TIO)) Approximately 6800H Approximately 1% Approximately 8% Approximately 0.5% 100% (exclusive use) |
| Number of tracks Music SFC | 14 18 tracks (works with the PCM driver) 1 track(s) |
| Remarks | A software envelope is provided as a PCM. Two types of software vibrato functions are provided as an FM/PSG. These resources and general sound sources with the same "look and feel". A drum kit can be created as an FM sound source. |

*If PCM is not used, the driver size decreases by approximately 1000H.

Activating the Sound Driver

The following procedures are used to control the V3.

- 1) Use a system call (macro) to initialize the Sound work space.
- 2) Call the starting address for the Sound Driver every V-INT (approximately 16 ms).

See the system call reference in this chapter for a method for requesting sound data.

Memory Map

The Sound Driver operates under the following memory map:

| |
|--|
| ROM memory map |
| Sound Driver (approximately 200H) |
| Sound Driver system unit |
| Data vectors (offset address) (4x4 bytes) |
| PCM data (undefined) |
| PSM data (undefined) |
| Mute data (undefined) |
| SE data (undefined) |
| Table envelope data (undefined, in units of 100H bytes) |
| Table volume data (undefined, in units of 100H bytes) |
| Register data (undefined, in units of 4 bytes) |
| FM sound source data (undefined, in units of 100H bytes) |

RAM Memory Map

| |
|--|
| Common work space (to PCM source with the number of PCM sources (undefined)) |
| Channel work space (to 4 channels for at maximum 80H) |
| VMUT 12 work buffer (200H bytes) |

Fill-In Data

The Driver is filled with the following information at a distance of 6CH from the starting address:

| Header | Type |
|--|--------------|
| Work space starting address | 4 bytes |
| Work space size | 4 bytes |
| PCM type (00H: none, 10H: linear, 20H: delta) | 1 byte |
| Number of sounds generated from the PCM sound source | 1 byte |
| Number of sounds generated from the FM sound source | 1 byte |
| Number of sounds generated from the PSG sound source | 1 byte |
| Number of sounds generated from the Pella sound source | 1 byte |
| Total number of tracks | 1 byte |
| Total number of tracks per music | 1 byte |
| Total number of tracks per SE | 1 byte |
| Driver version number | 4 bytes |
| Driver type | unidentified |

System Calls

The V3 uses system calls to support driver controls. This enables the V3 to accommodate driver version upgrades, as well as work-space changes, without requiring a modification of the V3 itself. Although it is possible to control the driver by directly rewriting the work space, as has been done in the past, to provide for future version upgrades it is recommended that the driver be controlled by means of system calls.

List of System Calls

| Number | Operation | Input register | Output register | Destroyed registers |
|--------|---|----------------|-----------------|---------------------|
| 00H | Initializes the Sound work space and the hardware's sound the Sound Driver. | D0 | None | None |
| 01H | Requests a piece of music. | D0 | None | 00H/0 |
| 02H | Requests an SE. | D0 | None | 00H/0 |
| 03H | Sets a fade-in/out. | D0 | None | None |
| 04H | Sets the music master volume. | D0 | None | None |
| 05H | Sets the SE master volume. | D0 | None | None |
| 06H | Sets the music master reverb size. | D0 | None | None |
| 07H | Sets the SE master reverb size. | D0 | None | None |
| 08H | Sets a piece. | None | None | None |
| 09H | Resets a piece. | None | None | None |
| 0AH | Writes communication data. | D0 | None | None |
| 0BH | Reads communication data. | None | D0 | None |
| 0CH | Requests to stop the music. | None | None | 00H/0 |

| | | | | |
|-----|---------------------------|------|------|------|
| 001 | Request to stop the SE | None | None | None |
|-----|---------------------------|------|------|------|

Rex Sabio
242

Details on System Calls

First, set the desired system call number in the register D0. Then, set register values as necessary, and call the address that is obtained by adding 8 to the starting address of the Sound Driver.

- 00H** **Initializing the Sound Driver and the hardware around the Sound Driver**
 Function: Initialize the Sound Driver and the hardware around the Sound Driver.
 Input: D0b = 00H
 D1b = 000:NTSC/PAL
 0 = sets to NTSC
 1 = sets to PAL
 Output:
 Destruction:
 Remarks:
- 01H** **Requesting a piece of music**
 Function: Performs music.
 Input: D0b = 00H
 D1b = piece number
 Output:
 Destruction: D0/A0
 Remarks: Because of a 4-byte request buffer, sounds can be produced in a maximum of four simultaneous interrupts (common to the SE multi-bus). Some request numbers can cause the SE to be performed (music numbers are not checked).
- 02H** **Requesting an SE**
 Function: Performs SE sounds.
 Input: D0b = 00H
 D1b = SE number
 Output:
 Destruction: D0/A0
 Remarks: Because of a 4-byte request buffer, sounds can be produced in a maximum of four simultaneous interrupts (common to the music piece numbers). Some request numbers can cause music to be performed (music numbers are not checked).
- 03H** **Fade-request**
 Function: Sets the fade-in/out option.
 Input: D0b = 00H
 D1w = sets the fade value.

| 16-bit | 7-bit |
|--------------|--------------|
| Fading depth | Fading speed |

Output
 Destruction
 Remarks

Fade-in operations can be performed by setting the fading depth to a negative number (2's complement).
 The allowable range of fading speed is 00H-7FH (in units of V-unit).

04H Setting the master music volume

Function: Sets the volume for the entire music.
 Input: D0b = 00H
 D1b = volume

Output:

Destruction
 Remarks: The allowable range of volume is 00H-7FH, where 00H indicates the maximum volume, and 7FH the mute option.

05H Setting the master SE volume

Function: Sets the volume for the entire SE.
 Input: D0b = 00H
 D1b = volume

Output:

Destruction
 Remarks: The allowable range of volume is 00H-7FH, where 00H indicates the maximum volume, and 7FH the mute option.

06H Setting the master/music transpose

Function: Transposes the entire music.
 Input: D0b = 00H
 D1b = transposition value

Output:

Destruction
 Remarks: A transposition value can be a negative value (2's complement).

07H Setting the master SE transpose

Function: Transposes the entire SE.
 Input: D0b = 00H
 D1b = transposition value

Output:

Destruction

| | |
|--------------------------------|---|
| Remarks: | A transposition value can be a negative value (2's complement) |
| 08H Requesting a pause | |
| Function: | Sets a pause |
| Input: | D0b ... 08H |
| Output: | |
| Destruction: | |
| Remarks: | Because PWM sound sources lack a function for stopping sound production on a channel-by-channel basis, once sound production is started it cannot be stopped. |
| 09H Resetting a pause | |
| Function: | Resets a pause. |
| Input: | D0b ... 09H |
| Output: | |
| Destruction: | |
| Remarks: | |
| 0AH Writing communication data | |
| Function: | Writes communication data into sound work space. |
| Input: | D0b ... 0AH D1b ... data |
| Output: | |
| Destruction: | |
| Remarks: | This is a processing action viewed from the main system. Currently data cannot be read on the Driver. Therefore, writing data from the main system will not alter the processing. |
| 0BH Reading communication data | |
| Function: | Reads communication data from sound work space. |
| Input: | D0b ... 0BH |
| Output: | D0b ... data |
| Destruction: | |
| Remarks: | This is a processing action viewed from the main system. |
| 0CH Requesting to stop music | |
| Function: | Terminates music. |
| Input: | D0b ... 0CH |
| Output: | |
| Destruction: | D0/A0/A2 |
| Remarks: | Because PWM sound sources lack a function for stopping sound production on a channel-by-channel |

basis, once sound production is started it cannot be stopped.

00H Requesting to stop an SE
Function: Terminates an SE.
Input: D0/b: 00H
Output:
Destination:
Remarks:

Because PWM sound sources lack a function for stopping sound production on a channel-by-channel basis, once sound production is started it cannot be stopped.

Rex Sabio
242

Data Request Numbers

For various pieces of data (music and SGs), the following range of request numbers can be specified:

| Music | SG |
|---------|---------|
| 00H-0FH | 01H-7FH |

The following effect commands can be requested:

| Number | Effect name | Effect |
|--------|-------------------------------|--|
| FDH | Fade-in | Produces a fade-in effect. |
| F0H | Fade-out | Produces a fade-out effect. |
| F2H | Mute | Stops the music being played. |
| F3H | Stopping SGs | Stops all SGs from which sounds are being produced. |
| F4H | Pausing | Pauses music. |
| F5H | Resuming a pause | Resumes music from a pause state. |
| F6H | Music master transposing up | Raises the music's master transposition by a halfnote. |
| F7H | Music master transposing down | Lowers the music's master transposition by a halfnote. |
| F8H | SG master transposing up | Raises the SG's master transposition by a halfnote. |
| F9H | SG master transposing down | Lowers the SG's master transposition by a halfnote. |
| FAH | Music master volume up | Increases the master volume for music by 1. |
| FBH | Music master volume down | Reduces the master volume for music by 1. |
| FDH | SG master volume up | Increases the master volume for the SG by 1. |
| FEH | SG master volume down | Reduces the master volume for the SG by 1. |
| FFH | Sound Driver initialization | Initializes the Sound Driver. |

Sound Data

This section describes the internal structure of sound data. For address specification, the term "address" refers to a relative address from the starting address shown in the following table.

Tap Vector

A tap vector stores an offset address of data (8 bytes per address).

| |
|---------------------------------------|
| Address of PCM information |
| Address of the PWM information table |
| Address of SE information |
| Address of the SE information table |
| Address of table envelope data |
| Address of table vibrato data |
| Address of FM rhythm list data |
| Address of FM sound source timer data |

Address Tables

An address table is contained in PCM, PWM, music, or in an SE, and in one place stores the addresses that point to those pieces of information (two volume addresses and also information in the case of PCM), tempo information and track addresses in the case of music).

PCM Data

PCM data stores addresses for PCM information. Any two-byte data relating to PCM is stored in terms of little-endian (LITTLE-Endian - UNKNOWN TERM).

Playback speed (simply a 280-weight value, not a sampling rate) (1 byte)

| |
|---------------------------------------|
| Parameter for pitch (1 byte) |
| Address of the data address (2 bytes) |
| Data size (2 bytes) |

Music Information Data

This data codes the information necessary for playing back a piece of music (tempo, address to sequence data).

| |
|---|
| NTSC/PAL tempo (in the order of NTSC and PAL) (2 x 2 bytes) |
| Address of PCM sound source/PCM sequence data (2 bytes) |

| |
|--|
| PCM sound source (1CH, master frequency (1 byte) |
| PCM sound source (5CH, master volume (1 byte) |
| (PCM sound source: 1CH, by driver selection) (4 bytes) |
| (FM sound sources 0-5 (FM PCM currently not available) (4x4 bytes) |
| PSG sound sources 0-3 (4x4 channels) (4x4 bytes) |
| PSG sound sources 0-3 (4x4 bytes) |

Rex Sabio
 242

SE Information

This field provides the data (number of required channels, etc.) necessary for producing SE sounds.

| |
|---|
| Number of required channels (1 byte) |
| Priority (1 byte) |
| Sound source ID for track 0 (1~52 bytes) |
| Sequence data address for track 0 (2 bytes) |
| Transition for track 0 (1 byte) |
| Volume level for track 0 (1 byte) |

FD See the section on "Identifying sound sources".

Table Vibrato/Table Envelope

These tables require a minimum size of 100H per table (word). Both data and commands are coded as numerical values. Vibrato data is expressed in 2s complements as 7FH (maximum positive number), 00H (neutral), and 80H (maximum negative number). Only the values 00H-7FH can be used in an envelope. Commands are expressed in 80H-FFH as indicated below:

| Number | Function |
|--------|---|
| 80H | Returns to the beginning of the table |
| #1H | Returns the last value |
| 82H | Moves to a specified table position |
| 83H | The phrased moves the sound to a specified table (see neutral position). The envelope stops the sound when the neutral position is reached. |

PM Drum Kit

The data sequence number is used as a key number during sequencing.

| Offset | Parameter name |
|--------|------------------|
| 00H | Timbre number |
| 01H | Volume |
| 02H | Musical interval |

| | |
|---------|----------------------|
| 00H | Pen-pal |
| 04H | Table vibrato number |
| 004-07H | System reserve |

FM Sound Source Parameters

FM sound source filter data is in a partially packed format so that it can be written directly into a register.

| Offset | Parameter name |
|--------|--|
| 00H | Connectivity/Feedback |
| 01H | Source/Multiple (bits 1, 3, 2, 4) |
| 02H | Key Scaling/Attack Rate (bits 1, 3, 2, 4) |
| 03H | AMDcay Rate (bits 1, 3, 2, 4) |
| 04H | Sustain Rate (bits 1, 3, 2, 4) |
| 11H | Sustain Level/Release Rate (bits 1, 3, 2, 4) |
| 12H | Total Level (bits 1, 3, 2, 4) |

Details of Sequence Commands

The following sequence commands can be used as music and SFX.

01H-7FH Tone length

Function: Sets the tone length. Given a musical interval, a sound can be produced solely on the basis of its length.
Remarks:

80H Pause code

Function: Sets the pause code. Suspends reading data for a specified interval of time.
Remarks:

81H-8CH Scale (C,G,A,D,D#,E,F,F#,G,GE,A#,B)

Function: Specifies a musical interval (eigh square range). Given a tone length, a sound can be produced solely on the basis of its musical interval.
Remarks:

CDH, data:b

Function: Writes communication data.
Remarks:

01H, number:b

Function: Requests an SFX.
Remarks:

G2H, offset:w, byte-count, data:b

Function: Writes a specified byte count to the offset for a specified sound workspace.
Remarks: This Sound Driver does not provide for malfunction that may occur as a result of using this command to rewrite the work space.

C3H, number:b

Function: Sets the FM sound source envelope to the SSC type.
Remarks: For details, see the "TM-3602 Application Manual".

04H, PMS/AMS data:b

Function: Sets a PMS/AMS.
Remarks:

05H, bank:b, register number:b, data:b

Function: Directly overwrites the register for an FM sound source.

Remarks For details, see the "Y&M-012 Application Manual".
 The Sound Driver does not provide for malfunction that may occur as a result of using this command to rewrite the work space.

C8H, mode b

Function

Changes FM source sound production methods.
 The 0 mode is the ordinary sound production mode; the non-0 mode is the DRUM mode.

Remarks

For a description of the DRUM mode, see the "Tone Editor Manual".

D0H-D7FH

Function

Sets the velocity.

Remarks

Velocities are converted as shown below and added to the sound volume level.

| Command number | D0H | D1H | D2H | D3H | D4H | D5H | D6H | D7FH |
|----------------|-----|-----|-----|-----|-----|-----|-----|------|
| Actual value | 30H | 20H | 30H | 30H | 30H | 30H | 30H | 30H |
| Command number | D8H | D9H | DAH | DBH | DOH | DOH | DOH | DOH |
| Actual value | 10H | 10H | 10H | 10H | 00H | 00H | 00H | 00H |

E0H, number b

Function

Changes timing and envelope.

Remarks

E1H, absolute-volume b

Function

Sets the absolute volume. The higher the numerical value, the smaller the volume.

Remarks

The absolute volume is specified in a 00H-7FH range for all sound sources.

E2H, relative-volume b

Function

Sets the relative volume. The higher the numerical value, the smaller the volume.

Remarks

The absolute volume is specified in a 00H-7FH range for all sound sources. The function does not check for an overflow.

E3H, point b

Function

Sets the pan-pot. If the Qsound is used on a PWM sound source, the command sets a Qsound point in MIDI-standard-based numerical values (30H-40H-7FH).

Remarks: This command is applicable only to FM and PV001 sound sources.

EAH, tune b

Function:

Sets the detune option in units of 1/32 half-tone.

Remarks:

ESH, delay b, stay b, increment b, limit b

Function:

After a sound is produced, waits for a time interval equal to a delay interrupt, adds the increment for each stay, and changes the sign when the limit is reached.

Remarks:



ESH, transpose b

Function:

Performs a transpose. Negative values (2's complements) can also be used in this command.

Remarks:

E7H, bend-value w

Function:

Performs a bend. Actually, the bending operation is the 16-bit version of the detuning operation.

Remarks:

Because of a 1-bit left shift that is performed internally, the actual resolution is 15 bits.

ESH, number b

Function:

Sets the table vibrato.

Remarks:

The 0 value specified in the number field resets the vibrato.

ESH, switch b

Function:

Temporarily mutes the vibrato. An 0 resets and a non-zero sets the vibrato.

Remarks:

This command is required in order to enable the commands ESH/ESH.

EAH, octave b
 Function: Sets the octave to an absolute value.
 Remarks:

EBH, octave b
 Function: Increases the octave by one.
 Remarks: Does not check for an overflow.

ECH, octave b
 Function: Decreases the octave by one.
 Remarks: Does not check for an overflow.

EEH, mode b (data -)
 Function: This command performs different operations, depending on the sound source selected, as follows:

| Sound source | Operation | Data type (oct) |
|-----------------------|--|-----------------|
| FM sound source (DCH) | Sets the sound effect mode. Specify the immediate value to be written to the register followed by a Mode#-Number (2 bytes per slot). | 2 bytes |
| FM sound source (DCH) | Allows you to select whether a value is to be used as a PCM or FM sound source. Specify the immediate value to be written to the register. | 1 byte |
| PCM sound source | Sets a loop. Specify the immediate value to be written to the register. | 1 byte |
| PWM sound source | Enables the Closed. An 0 turns the Closed off, a non0 value turns it on. | 1 byte |

Remarks: For details, see the respective sound source manuals.

PDH, NTSCdiv, Pdiv, or
 Function: Sets the tempo.
 Remarks: "NTSC" stores the value determined according to the following formula:

$$((\text{tempo}/150) \times 5100 + (\text{remainder of tempo}/150) \times 5100/150) \times 5$$

 The value of Pdiv is NTSC value x 5/5.
 This command is not applicable to the SE (the tempo is fixed at 150).

FIH
 Function: Disables turning the next note off.

| | |
|---|---|
| Remarks: | |
| F0H, gate, value:b Function: Remarks: | Sets the gate. |
| F3H, address:w Function: Remarks: | Saves the current address in a work space and moves it to a specified position. An address (including a header) is a relative value from the beginning of music data. |
| F4H Function: Remarks: | Moves a given address to a stored address, discarding the stored address. Fetches addresses on a first-in, first-out basis. |
| F5H-F7H, count:b Function: Remarks: | Moves to an address, specified by F5H-F7H, a specified number of times. This is a repeat function. There are three commands of this type. Consequently, a maximum of three nesting levels can be used. |
| F8H-FAH Function: Remarks: | Terminates the repeat count specified by F5H-F7H, and moves to the data following F8H-F7H. |
| FBH-FDH Function: Remarks: | Saves the starting repeat pointer. This command is meaningful only if used in conjunction with the commands F5H-F7H. |
| FEH Function: Remarks: | Saves the current address. This command is meaningful only if 1 is specified in F5H. |
| FFH, mode:b Function: Remarks: | This command indicates the end of track data. If a non-0 mode is in effect, control moves to the data following FEH. In the case of a non-0 mode, use FEH to set a move-to point. |

* This data is applicable only to the Devcon. Not all data can be controlled by means of data that is based on the MIDI converter.

Rex Sabio
242

Identifying Sound Sources

The following IDs are assigned to various sound sources

| Sound Source | PCM | PWM | PSG1 | PSG2 | PSG3 | PSG4 | PSG5 | PSG6 | PSG7 | PSG8 | PSG9 |
|--------------|------|------|------|------|------|------|------|------|------|------|------|
| ID | 00H | 01H | 02H | 03H | 04H | 05H | 06H | 07H | 08H | 09H | 0AH |
| Sound Source | PCM0 | PCM1 | PWM0 | PWM1 | PWM2 | PWM3 | | | | | |
| ID | 10H | 11H | 0BH | 0CH | 0DH | 0EH | | | | | |

Notes

The following ranges of data can be used as pseudo-VB:

| Item | Range | Remarks |
|----------------------|---------|--|
| Key | 00H-7FH | For note data keys, the octave and the scale are set separately. Keys are defined according to MIDI standards. Although it is possible to implement an instrument (e.g., organ) 1 and 2 cannot be played correctly. |
| Volume | 00H-7FH | The maximum allowable volume is 00H, the minimum 7FH. The sound sources are implemented according to their hardware specifications without the influence of volume levels. In user-defined volumes, other than the FM sound sources, a specified volume level cannot be produced because of hardware/software limitations. |
| Timbre | 00H-7FH | The timbre parameter is not applicable to PCM or PWM. For PSG, a timbre is treated as a software envelope number. |
| Pan-pot | 00H-7FH | The allowable range of numeric values for the pan-pot category is based on MIDI standards. This range is not applicable to sound sources (i.e., PCM and PSG) that do not have a pan-pot due to hardware/software limitations. |
| Sound Driver address | | Any address that is executable by the 68000 is allowed. |
| Sound work space | FF0000H | This is fixed as a general rule, and cannot be modified. |

| | | |
|-----------------------------------|-------|--|
| Maximum number of sounds produced | 15-16 | This number varies with the particular PCM driver selected. |
| Number of SE tracks | 5 | This is fixed as a general rule, and cannot be modified. |
| Switching between FM5 and DSA | - | Switching between FM5 and DSA can be performed either by not entering any data at the FM5 or by using the command ESC . |
| YMS12 write-in data buffer | 200H | This is fixed as a general rule and cannot be modified. |

Rex Sabio
 242